Passerelle à base de processeurs réseaux pour le contrôle des flux de grille

Sébastien Soudan* et Pascale Primet*
*Équipe RESO, Laboratoire de l'Informatique du Parrallélisme
UMR CNRS-ENS Lyon-UCB Lyon-INRIA 5668
École Normale Supérieure, 46, Allée d'Italie
69364 Lyon CEDEX 07, France
Email : prenom.nom@ens-lyon.fr

***Abstract*— Les grilles de calcul hautes performances sont aujourd'hui constituées de grappes d'ordinateurs interconnectées par des réseaux de cœur sur-dimensionnés. Cependant les liens d'accès dont les débits restent inférieurs aux débits disponibles au sein d'une grappe, constituent de sévères goulets d'étranglement. Nous proposons d'optimiser les performances globales des grilles par une organisation et un contrôle des transferts permettant à l'ordonnanceur de tâches de planifier les communications en même temps que les tâches. En se basant sur un nouveau modèle de réservation de ressource réseau dans la grille, cet article présente l'architecture d'équipements en charge de l'allocation et du contrôle de la ressource réseau à l'interface LAN/WAN. Pour implanter cette solution tout en conservant les débits d'accés (1 Gb/s), nous avons développé un prototype basé sur la technologie des *Network Processors IXP2400* d'INTEL. Nous discutons ici les principaux choix conceptuels et quelques résultats expérimentaux.**

I. INTRODUCTION

La formidable croissance de la puissance des processeurs prédite par la loi de Moore a fortement influencé l'évolution du domaine du calcul au cours des 40 dernières années. Les avancées encore plus rapides des technologies de transmission optique sont à même de produire une nouvelle révolution non seulement dans le domaine des télécommunications mais aussi dans le domaine du calcul. Les grilles peuvent répondre aux besoins de calcul croissants des grandes applications scientifiques (de biologie, d'astro-physique, de physique des hautes énergies, d'imagerie médicale) et industrielles (simulation numérique) [1], [2]. Pour créer cette agrégation complexe de ressources, il était naturel de se baser sur la technologie IP de l'Internet qui répond aux exigences d'interopérabilité, masque l'hétérogénéité des équipements et des technologies et présente d'excellentes propriétés de robustesse et de résistance au facteur d'échelle. Les applications réparties sur une grille utilisent les protocoles TCP-UDP de l'Internet qui sont largement diffusés et disponibles sur les noeuds de calcul et permettent de communiquer sur les liens longue distance.

Cependant, on observe que dans le cadre des grilles aujourd'hui déployées, les capacités sans cesse croissantes et les services de plus en plus sophistiqués fournis par les infrastructures réseau ne sont pas accessibles aux applications. Le nuage réseau global présente un mode de partage non contrôlé globalement [3] qui lui confère des performances, une sécurité et des fonctionnalités très variables et parfois incompatibles avec les algorithmes de calcul classiques. En conséquence, la performance d'exécution des applications, en grande partie déterminée par les mouvements de données, est souvent délicate à prédire et à obtenir en pratique. La possibilité de perdre des messages en rafale, les délais de transmission importants et variables, le comportement très dynamique des liens entre processus distants remettent profondément en question les modèles et les techniques développées dans le contexte du calcul parallèle et du calcul distribué sur grappe de calcul. Il est pourtant fondamental de s'assurer que le potentiel considérable offert par l'agrégation de ressources ne soit pas inexploité voire gâché par l'inadéquation des modèles et des mécanismes de communication mis en oeuvre.

Le besoin de qualités et de différentiation de service est très important [4]. Dans une grille, il faut faire cohabiter des flux aux contraintes antagonistes tels que les flux de contrôle, les barrières de synchronisation et les transferts massifs de données. Le modèle de service unique *Best Effort* pour l'acheminement de paquets et l'omniprésent protocole de transfert TCP offrant une fiabilité et un ordre total, un partage équitable de la bande passante sans aucun contrôle des délais présente de sérieuses limitations ici. Le critère global d'équité (*max-min fairness*) classiquement recherché pour le partage de la bande passante n'est pas adapté à ces nouveaux usages et à des objectifs d'optimisation différents. Ici, la réservation de bande passante s'impose.

Cet article s'intéresse à ce problème particulier du contrôle des flux haut débit dans le cadre d'un modèle d'*overlay* de grille. La section suivante explicite notre modèle de contrôle des flux. La section 3 présente l'architecture d'un équipement baptisé *grid gateway* que nous proposons pour l'exécution de l'algorithme distribué d'allocation de bande passante et le contrôle en ligne des trafics et des flux traversant le lien d'accès. Nous présentons un prototype de cet équipement basé sur la technologie des processeurs réseau (*network processor*) que nous avons développé. Les résultats expérimentaux sont donnés en section 4. Enfin, la section 5 replace ces travaux dans l'état de l'art. La conclusion et les perspectives sont développées en section 6.

II. Réservation de bande passante et contrôle des flux de la grille

Dans une grille basée sur des réseaux TCP/IP, il est difficile de prévoir à l'avance la durée d'une tâche car on ne sait pas quand les données nécessaires seront disponibles sur les ressources de calcul. Dans un réseau filaire, pour qu'un transfert soit déterministe, il ne doit pas être perturbé par d'autres flux. C'est le principe de fonctionnement des réseaux locaux haute performance des grappes de calcul comme Myrinet où un circuit éphémère (*wormhole*) est alloué de bout en bout pour un transfert. Ainsi, les flux à faible latence des applications parallèles (MPI) utilisent abondamment ces garanties strictes que seule une forme plus ou moins virtuelle de réservation de ressource peut offrir.

Est-ce que cette approche peut s'étendre aux communications longues distances ? Dans [5] nous avons proposé et étudié un nouveau modèle de réservation de bande passante adapté aux flux et aux réseaux très haut débit des grilles et avons proposé des algorithmes d'ordonnancement associés. Cette proposition se base sur un certain nombre d'observations : les grilles connectent un faible nombre de ressources interagissant (e.g. 10^3 et non 10^8 comme dans l'Internet), la capacité d'une seule source (1Gb/s) est comparable à la capacité des goulets d'étranglement (1 ou 10Gb/s), les délais entre sites sont importants, les transferts sont massifs, le nombre de flux actifs tend à être faible et les flux tendent à être très longs (heures, jours). Ceci élimine le problème de passage à l'échelle aussi bien en nombre de flux qu'en nombre d'états à mettre à jour qui freine le déploiement des schémas de réservation dans Internet (cf IntServ). Il a été observé que dans une grille, si un contrôle d'admission pro-actif n'est pas appliqué, les performances se détériorent rapidement, à cause du profil "en rafale" des demandes, même lorsque la charge est moyenne voire faible (moins de 50%). Nous ne détaillons pas ici le modèle ni les fonctions objectifs, ni les heuristiques proposés, mais en donnons les principes généraux.

L'idée sous jacente est de transposer l'approche adoptée dans les réseaux de grappes à des réseaux longue distance. Le but est d'isoler les flux pour qu'ils se comportent comme des flux seuls dans des liens bien approvisionnés et ainsi qu'ils ne subissent que des pertes liées à leurs comportements. La fonction d'allocation prise en charge par les interfaces des réseaux rapides de grappes est ici déportée vers les passerelles d'accès, qui, dans notre modèle, constituent les seuls points d'engorgement. Nous imposons à chaque flux une limite de débit de telle sorte que la somme de ces débits soit inférieure aux bandes passantes des liens rencontrés. Les flux traversant plusieurs passerelles, les limites de débit doivent également être négociées avec les autres passerelles. Nous ajoutons un aspect temporel à l'approche de réservation de bande passante au niveau des passerelles. En effet, les volumes à transférer sont supposés connus à l'avance. Les sources peuvent donc être autorisées à émettre pendant une fenêtre temporelle donnée comme le montre la Figure 1. De ce fait chaque passerelle connaît par avance son "planning" d'utilisation et peut donc accepter (de son point de vue) un nouveau flux ou le rejeter. La mise en accord des passerelles d'accès pour planifier un flux de bout en bout se fait par l'échange de leurs ordonnancements respectifs et la négociation.

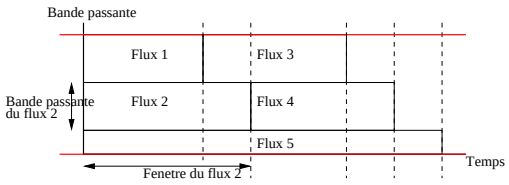

Fig. 1. "Planning" de flux

Ce modèle s'intéresse en particulier aux flux massifs et se base sur un modèle particulier de réseau avec un cœur virtuellement complètement maillé. Nous travaillons actuellement la généralisation de ce modèle théorique pour y intégrer les flux faibles latence et les autres flux *best effort* ainsi qu'un modèle de réseau hiérarchique plus complet. Ce modèle peut être mis en œuvre par un *overlay* de contrôle dont l'objectif est d'offrir un service d'ordonnancement, de gestion et de contrôle des flux de grille aux points d'engorgement potentiels. Ce service peut coopérer avec le système global de gestion des ressources ainsi qu'avec les processus d'applications.

Le contrôle des flux au point d'accès consiste à vérifier que chaque flux respecte le débit ainsi que la fenêtre temporelle allouée. Dans la suite de cet article, nous nous intéressons à la description et à l'implantation d'une architecture de passerelle de grille, élément actif de l'*overlay* de grille. Chaque passerelle de grille est localisée à l'interface entre le réseau local et le réseau longue distance. Elle est traversée par tous les flux sortants ou entrants dans le site (grappe) associé.

III. Conception d'une passerelle à base de *Network Processors*

L'architecture de la passerelle comporte deux plans fonctionnels : le plan données et le plan contrôle. Le rôle du premier est de transmettre les paquets alors que le second contient l'"intelligence" de la passerelle. Les passerelles sont classiquement [6] vues comme une succession de blocs fonctionnels reliés par des files d'attentes. Ces blocs sont : *réception (Rx)*, *Routage + contrôle des flux* et *transmission (Tx)*. Le bloc *Rx* récupère les données arrivant sur les interfaces et rassemble les paquets en mémoire. Les files d'attentes ne contiennent que des descripteurs référençants les données en mémoire et des informations sur les traitements réalisés et ceux à effectuer. Le bloc représenté sur la Figure 2 effectue le routage puis le contrôle de flux. Il s'agit d'identifier le flux auquel appartient le paquet, de vérifier qu'il n'est pas en dehors du profil puis d'autoriser ou de rejeter le paquet. Cette étape utilise une table de flux autorisés maintenue par le plan contrôle. Enfin le bloc *Tx* est chargé d'envoyer les paquets sur l'interface.

Le bloc de routage et contrôle de flux est le seul qui nécessite des informations provenant du plan contrôle. Ce dernier, représenté sur la Figure 3, est responsable des choix

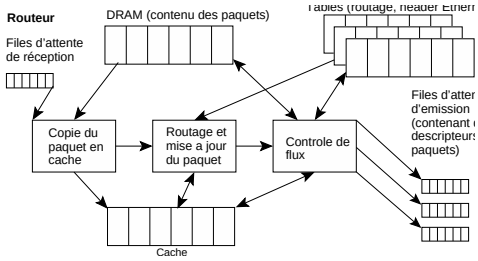

Fig. 2. Bloc de routage et contrôle de flux

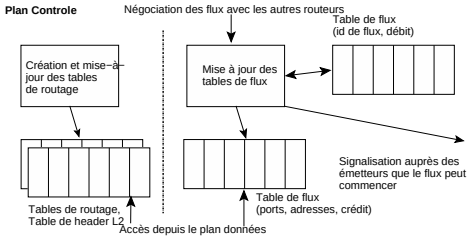

Fig. 3. Plan Contrôle

et du maintien à jour des règles de routage et de contrôle de flux. Nous utilisons deux tables pour effectuer ce contrôle. La première contient une description des flux (protocole, ports et adresses source et destination) ainsi qu'un compteur de crédit de bande passante (en paquets). Celui-ci est décrémenté par le plan données lorsqu'un paquet correspond au flux décrit. Si il atteint 0, les paquets sont rejetés. Les compteurs sont crédités par le plan contrôle à intervalle régulié en fonction de la seconde table qui contient la description du profil de trafic accordé à un flux (débit en paquets par seconde, date de début, date de fin). Cette seconde table permet de réaliser simplement la limitation dans le temps, le compteur de crédit n'est incrémenté que dans la fenêtre temporelle du flux.

Pour évaluer la faisabilité et les performances d'une telle architecture, nous avons implanté un prototype à l'aide de l'architecture de processeur réseau *IXP2400* d'INTEL. Ces plateformes disposent d'un nombre important de processeurs, de trois interfaces Gb/s pour la version *ENP-2611* de RADISYS, ainsi que d'une quantité assez importante de mémoire. Elles sont equipées d'un *Xscale* à 600Mhz sur lequel tourne LINUX ainsi que 8 MicroEngines (*ME*) qui sont des processeurs dédiés au traitement de paquets pouvant gérer 8 *threads* chacun.

Nous utilisons sept *ME* : un pour *Rx*, quatre pour *Routage*, et deux pour *Tx*. Le code du bloc de routage fonctionne sur quatre *ME* (soit 32 *threads*). Un *thread* est responsable du traitement d'un paquet à la fois. La fonction qui vérifie le profil d'un flux a besoin du quadruplet identifiant le flux auquel appartient un paquet. Chaque entrée dans la table de crédit fait 20 octets. Nous sommes dans un contexte de grille, nous supposons donc que le nombre de flux simultanés ne dépasse pas quelques milliers. La taille de la table n'est donc pas trop importante. Notons que nous avons une classe de flux particulière utilisant la première entrée de la table. Celle-ci est utilisée pour tous les paquets qui n'ont pas de flux décrits. Afin de conserver l'ordre des paquets, les *threads* des *ME* de routage sont ordonnés grâce à un signal propagé de *thread* en *thread*. Les blocs *Rx* et *Tx* proviennent du SDK RADISYS. Chacune des deux parties du plan contrôle est implantée sur le XSCALE.

IV. EXPÉRIMENTATIONS

Les tests suivants ont été réalisés sur des machines biprocesseur P4 Xeon 2.8GHz avec des noyaux LINUX 2.6.12.5.

L'application *StaticFwd* fait partie du SDK RADISYS. Dans sa version originale, elle retransmet les paquets entrants par la première interface ressortent par la seconde, ceux entrant par la seconde ressortent par la troisième… et ce sans modification. Cette application utilise trois *ME*, un pour *Rx*, un pour faire le "routage" et un pour *Tx*. Nous avons modifié ce code pour qu'il n'utilise que les deux premières interfaces et transmette les paquets entre celles-ci afin d'évaluer les performances de l'*IXP2400* avec le minimum de traitements. Nous avons utilisé *Iperf* pour générer le trafic. Nous pouvons ainsi mesurer le débit en *full-duplex* que nous pouvons espérer obtenir entre deux machines avec l'*IXP* au milieu. Les débits obtenus sont de 793+791 Mb/s en TCP et 819+806 Mb/s en UDP alors qu'en *back-to-back*, nous obtenons en TCP 890+903 Mb/s et 860+821 Mb/s en UDP. Lorsque les machines sont reliées par l'intermédiaire du switch, nous avons 863+863 Mb/s en TCP et 808+873 Mb/s en UDP. Les performances sont donc un peu inférieures (10%) lorsque l'on passe par l'*IXP* avec l'application *StaticFwd*. Comparons maintenant ces résultats avec ceux obtenus sur la passerelle avec gestion des flux.

Pour cela, nous nous plaçons dans le pire des cas du point de vue de la recherche de flux, quand il n'y a pas d'entrée spécifique pour le flux considéré et que celui-ci est comptabilisé avec la classe par défaut qui est utilisée lorsque la recherche échoue. Afin de saturer les liens, nous génerons le trafic en anneau, chacune des machines est configurée pour recevoir le trafic de la machine précédente et pour émettre vers la machine suivante. Les résultats en UDP que nous avons obtenu sont 793+807+876 Mb/s. Il n'y a donc pas de perte importante de performance par rapport à ce que nous obtenons entre deux machines avec un switch.

Pour tester le contrôle de flux, nous utilisons un trafic *half-duplex* entre deux machines et lui imposons différentes limites de débits sur différents intervalles temporels. Nous remarquons sur la Figure 4 que le débit n'est pas très lisse. Cela est dû au fait que nous mettons à jour le crédit de paquets tous les dixièmes de seconde. Le temps caractéristique de TCP est lié au RTT (ici 0.2 ms) donc le débit a le temps de croître du fait du mécanisme de contrôle de congestion jusqu'à ce que le crédit soit épuisé.

Dernière évaluation, nous effectuons les mêmes tests que pour *StaticFwd* et qu'en *back-to-back*. Nous utilisons deux machines reliées par notre passerelle et obtenons 849+833 Mb/s en TCP et 840+857 Mb/s en UDP. Ces résultats sont meilleurs que ceux obtenus avec *StaticFwd* alors que le code est plus complexe mais nous utilisons plus de *ME*. Notre passerelle peut gérer les trois interfaces à la fois avec des débits proches

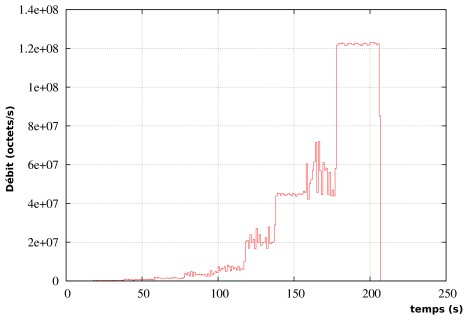

Fig. 4. Débit limité par le passerelle

des maximaux. Par ailleurs, l'ensemble des contrôles se fait sur la carte, l'ajout de l'algorithme d'ordonnancement sur le processeur central sera donc sans influence sur les performances.

V. Travaux relatifs

Tandis que le partage des ressources de calcul et de stockage ont été largement étudiées [7], l'idée d'intégrer la gestion de la ressource réseau dans les environnements de grille gagne de l'attention. Par exemple, la réservation de ressource réseau a été investiguée dans le contexte des grilles par [4]. L'architecture de réservation et d'allocation de ressource de Globus (GARA) proposée, introduit l'idée des réservations à l'avance et de la gestion de bout en bout de la QoS pour différents types de ressources (bande passante, stockage, calcul). Les algorithmes et les méthodes proposées sont très classiques et se basent sur les principes Intserv et DiffServ . Dans les faits cette proposition n'est pas encore déployée de manière très intensive dans les grilles car les réseaux longues distances n'offrent pas les services requis de bout en bout. Notre approche est beaucoup plus souple dans la mesure où elle se base sur une topologie réseau particulière quoique réaliste sans goulet d'étranglement dans le coeur et s'intéresse plus particulièrement au domaine privé qui présente les liens les plus congestionnés. Le groupe *Grid High-Performance Networking* (GHPN) du *Open Grid Forum* (OGF) spécifie actuellement le concept et les interfaces de service réseau de grille (*grid network service*). Les travaux présentés ici s'intègrent parfaitement dans ce contexte. Le problème du partage optimal de bande passante a aussi été étudié dans les grilles par [8]. Le modèle de réseau et le modèle de réservation adoptés sont différents de ceux proposés dans cet article. Par ailleurs, dans le cadre de projet ambitieux d'utilisation flexible des réseaux optiques dans les grilles [9], la problématique de réservation de bande passante dans les mondes optiques et Ethernet est aussi étudiée. Dans le prolongement des études développées dans cet article, nous étudions comment adapter notre modèle à ce contexte d'interconnexion avec les réseaux optiques contrôlables par l'utilisateur.

Quelques projets d'équipements d'extrémité ou de bordure ont été réalisés à base de *Network Processors*, on peut noter [10]. Les architectures *IXP* ont été utilisées avec succès pour réaliser des fonctions d'émulation de délai, d'espacement de paquets. En ce qui concerne la limitation de débit du coté des hôtes, qui est une des piéces de l'architecture proposée, [11] propose un mécanisme de limitation et de lissage par flux.

VI. Conclusion

Nous avons présenté un modèle d'*overlay* de contrôle des flux de grille permettant d'accroitre le déterminisme des performances des communications dans une grille en introduisant des mécanismes de réservation de bande passante. Pour implanter cette solution, nous avons décrit une architecture d'équipement de type passerelle de grille qui doit supporter des traitements à haut débit en utilisant des *IXP2400*.

Les *IXP* offrent de nombreuses solutions quant à la disposition des blocs fonctionnels et offre un très haut niveau de performances, cependant le développement dans ce type d'environnement s'avère particulièrement ardu du fait de la multiplicité des processeurs, architectures et mémoires.

Notre passerelle supporte le Gb/s mais ne dispose pas de *slow path* pour les paquets qui nécessitent plus de traitement comme le besoin d'effectuer une requête ARP.

Nos perspectives sont l'implantation de l'algorithme d'ordonnancement des flux, le déploiement de cette solution dans le contexte de la plate-forme Grid5000, l'extension du modèle au cas où le réseau de cœur n'est pas considéré comme surdimensionné et au cas où les congestions dans le réseau local sont prises en compte ainsi que le cas où le réseau dispose d'un plan contrôle unifié de bout en bout (comme GMPLS).

References

[1] F. Berman, G. Fox, and A. J. Hey, *Grid Computing : Making The Global Infrastructure a Reality*, S. in Communications Networking and D. Systems, Eds. Wiley, 2003, iSBN : 0-470-85319-0.

[2] Open Grid Forum, "Page web," 2005, http://www.ogf.org. [Online]. Available : http://www.ogf.org

[3] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Transaction on Networking*, vol. 3, pp. 365–386, Aug. 1995.

[4] I. T. Foster, M. Fidler, A. Roy, V. Sander, and L. Winkler, "End-to-end quality of service for high-end applications," *Computer Communications*, vol. 27, no. 14, pp. 1375–1388, 2004.

[5] L. Marchal, P. Primet, Y. Robert, and J. Zeng, "Optimizing network resource sharing in grids," in *Proceedings of the Intl. Conference IEEE GLOBECOM'05, USA*, Nov. 2005.

[6] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, August 2000.

[7] K. Czajowski, I. Foster, and C. Kesselman, "Resource co-allocation in computational grids," in *Proc. IEEE the eighth International Symposium on High Performance Distributed Computing*, Aug. 1999, pp. 219–228.

[8] L. Burchard, H.-U. Heiss, and C. A. F. De Rose, "Performance issues of bandwidth reservations for grid computing," in *Proc. IEEE the 15th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'03)*, Nov. 2003, pp. 82–90.

[9] OptIPuter, "Page web," 2005, http://www.optiputer.net/. [Online]. Available : http://www.optiputer.net/

[10] K. Nakauchi and K. Kobayashi, "Studying congestion control with explicit router feedback using hardware-based network emulator," in *In Proceedings of the International workshop on Protocols for Long Distance Networks (PFLDNET 2005)*, Lyon, France, 2005.

[11] R. Takano, T. Kudoh, Y. Kodama, M. Matsuda, H. Tezuka, and Y.Ishikawa, "Design and evaluation of precise software pacing mechanisms for fast long-distance networks," in *In Proceedings of the International workshop on Protocols for Long Distance Networks (PFLDNET 2005)*, Lyon, France, 2005.